

## **BAB III**

### **ANALISA DAN PERANCANGAN SISTEM**

Pada bab ini akan membahas tentang analisa dan perancangan sistem. Analisa sistem meliputi Analisa masalah, deskripsi produk, analisa kebutuhan, kehandalan dan analisa ekonomi. Sedangkan perancangan sistem meliputi arsitektur sistem, perancangan komunikasi, *scheduling system*, *report*, *packaging*, desain *database*, perancangan *user interface*.

#### **3.1 Analisa Sistem**

Sebagaimana yang telah dijelaskan pada Bab 1 Pendahuluan, sistem yang dibangun adalah sistem yang berbasis *web application*. Sedangkan *software* yang dibangun hanya pada bagian *server* saja. Bagian ini merupakan bagian *central* dari *smart home clients* (Door Lock, Lamp Control, Watt Meter & Electricity Control, Camera Watcher dan IR Custom) sehingga menjadi pusat dari penyimpanan data-data sensor dan menjadi komando untuk *clients* yang aktif. Selain itu bagian ini juga merupakan bagian yang akan berinteraksi langsung dengan *user*, sehingga membutuhkan *user interface* yang nyaman dan mudah digunakan oleh *user*.

##### **3.1.1 Analisa Masalah**

Rumah merupakan salah satu kebutuhan primer manusia yang biasa disebut papan. Membangun rumah yang nyaman membutuhkan biaya yang tinggi, belum lagi biaya perawatan, keamanan dan lain sebagainya. Sering kali terganggu dengan pekerjaan rutin seperti menyalakan lampu pada sore hari, mematikan lampu pada jam tidur, lupa mematikan lampu ketika bepergian, terlebih lupa mengunci pintu pada malam hari. Dengan permasalahan ini membutuhkan sebuah otomatisasi untuk menggantikan pekerjaan tersebut.

Solusi sederhana yang biasa dipilih oleh pemilik rumah biasanya adalah menyewa pembantu rumah tangga dan penjaga rumah untuk membantu mengelola rumah dan mengamankan rumah. Solusi ini tidak murah, bahkan tidak dapat diterapkan pada keluarga menengah kebawah yang mempunyai penghasilan yang tidak besar. Solusi lain untuk menyelesaikan permasalahan ini adalah dengan membangun sebuah perangkat cerdas pembantu rumah tangga yang bisa diakses dari dalam maupun luar rumah (internet) yaitu seperti *smart home*.

### 3.1.2 Deskripsi Produk

Sistem yang akan dibangun berupa *web*. Pada *web application* terdapat sebuah Api yang nantinya akan digunakan oleh *client device* seperti Door Lock, Lamp Control, Watt Meter & Electricity Control, Camera Watcher dan IR Custom sebagai media komunikasi dengan *server*.

Sistem ini didesain untuk *multi-user*, sehingga tidak hanya satu akun *user* saja yang akan digunakan karena dalam sebuah rumah biasanya terdapat banyak orang yang nantinya akan menjadi pengguna pada sistem *smart home* ini. Tetapi terdapat satu akun pengguna yang tingkatnya paling tinggi yaitu *admin*. Akun *admin* bisa melakukan semua fitur yang dimiliki sistem termasuk manajemen pengguna lain.

Pada *web application* pengguna dapat melakukan kontrol, pengaturan pengguna, pemberian jadwal sistem dan semua kontrol yang ditawarkan pada sistem ini.. Seperti menyalakan dan mematikan lampu, meremote *IR Custom* (TV, AC dsb.), *monitoring* Watt Meter, mengunci pintu dan *monitoring* kamera. Bergantung izin penggunaan *device client* yang diberikan oleh *admin*.

### 3.1.3 Analisa Kebutuhan

Pada perancangan alat ini karena menggunakan perangkat lunak dan keras untuk membuatnya maka dapat dianalisa kebutuhan apa saja dari perangkat lunak dan keras yang dibutuhkan untuk membuatnya.

Perangkat lunak yang digunakan adalah :

1. Sistem Operasi Raspbian berikut paket-paketnya :  
apache2, mysql-server, php5-mysql, phpmyadmin
2. *Web Editor* seperti Notepad++, PhpStorm atau *editor* lainnya.
3. *Web Browser*, digunakan untuk menampilkan dan mengakses website.
4. WinSCP, digunakan untuk melakukan remote akses ftp pada *server* Raspberry Pi 3.
5. Putty, digunakan untuk *remote console* pada *server* Raspberry Pi 3.
6. Framework CI 3 dengan tambahan *Modular Extensions*.
7. *Template* gratis dari Admin LTE beserta modul-modul pendukungnya
8. Ngrok, sebagai aplikasi *tunneling* sehingga aplikasi bisa di-remote di *internet*.

Perangkat Keras dan kebutuhan pendukung yang digunakan :

1. Raspberry PI 3, sebagai *server*
2. Internet untuk akses online

#### **3.1.4 Kehandalan**

Kehandalah adalah bagian daya tarik dari sebuah produk dan menjadi tujuan dari konsumen untuk memilikinya. Berikut kehandalan yang akan dibuat :

1. Sinkronisasi *device client* dan *server* yang mudah.
2. Perangkat keras yang modular sehingga pengguna bisa menyesuaikan kebutuhan yang ada di rumah.
3. Sistem akan berjalan kembali dengan stabil ketika terjadi mati listrik.
4. Sistem bisa diakses dari luar rumah menggunakan internet.
5. Semua koneksi dengan *device* menggunakan jaringan nirkabel sehingga memudahkan pemasangan.
6. Sistem berjalan di banyak *platform* seperti windows, linux, android dan ios.
7. *Server* yang dirancang berjalan di mini pc sehingga lebih hemat energi.
8. Terdapat otomatisasi *device* seperti fungsi “Going Out”, “Going Home”, “Sleep” dan “Wake Up”.
9. *Multi User*, sehingga bisa membuat banyak *user* dan diberikan otorisasi berbeda pada setiap *user*.
10. Memberikan otomatisasi kepada *device client* seperti mematikan lampu sesuai dengan jadwal.

#### **3.1.5 Analisa Ekonomi**

Sistem ini ditujukan untuk masyarakat menengah keatas. Konsep modularitas pada sistem membuat pengguna tidak diwajibkan untuk memiliki semua perangkat, jadi hanya *device* yang dibutuhkan yang harus dimiliki.

Penggunaan sistem ini tidak hanya untuk memberikan kenyamanan dan keamanan pada pengguna, otomatisasi sistem memberikan efisiensi pemakaian perangkat yang berlebihan sehingga mengurangi terjadi pemborosan dalam pemakaian sumber daya listrik.

### **3.2 Perancangan Sistem**

Pada perancangan sistem akan dijelaskan bagaimana perangkat lunak dan keras dibuat dan bisa saling terkoneksi antara satu dengan yang lain. Pada

perancangan ini akan menjelaskan bagaimana arsitektur yang dibangun dan bagaimana *device clients* bisa berkomunikasi dengan *server* dan membuat *server* lokal bisa diakses dari luar menggunakan *internet*..

### 3.2.1 Arsitektur Sistem

Adapun desain arsitektur dari sistem yang akan dibuat dapat dilihat pada Gambar 3.1.

1. Modul Layer	Door Lock	Lamp Control	Watt Meter & Electricity Control	Camera Watcher	IR Custom
1. Communication Media Layer	Wireless Media Communication (Wifi)				
3. Server Layer	Server Local		Server Public		
4. Platform Layer	Web				
5. User Communication Media Layer	Wifi		Internet		
6. UI Layer	Browser				

**Gambar 3.1** Arsitektur Jaringan

Pada gambar 3.1 komponen arsitektur sistem dapat dikelompokkan menjadi beberapa *layer*, sebagai berikut :

- 1) *Modul layer* : merupakan *layer* paling dasar dari keseluruhan sistem. *Layer* ini berhubungan langsung dengan sensor-sensor dan atau alat-alat yang akan dikontrol.
  - a. Door Lock : Modul ini berfungsi sebagai alat pengunci pintu secara digital yang dapat dikontrol oleh *server*.
  - b. Lamp Control : Modul ini berfungsi untuk mematikan dan menyalakan lampu, bahkan dapat membuat lampu redup dan terang menyesuaikan dengan keinginan pengguna.
  - c. Watt Meter & Electricity Control : Modul ini berfungsi untuk melakukan monitoring arus, informasi yang didapat adalah penggunaan listrik per jam

- menyesuaikan keinginan pengguna, serta dapat memutuskan atau menyambungkan arus yang dikontrol.
- d. Camera Watcher : Modul ini merupakan modul produk, yaitu IP Camera. Sehingga tidak perlu dibuat. Fungsi dari produk ini sebagai monitoring ruangan dan melakukan snapshot video.
  - e. IR Custom : Modul ini merupakan modul penghubung semua alat rumah tangga yang terdapat remote control yang menggunakan media IR. Sehingga semua perangkat rumah tangga yang terdapat remote control IR dan terpasang oleh modul ini dapat dikontrol oleh sistem.
- 2) *Communication Media Layer* : merupakan *layer* kedua yang menghubungkan antara modul *layer* dan *server layer*. Pada arsitektur ini sistem menggunakan media jaringan tanpa kabel seperti *wifi* sebagai media komunikasi antara modul *layer* dan *server layer*.
- 3) *Server Layer* : merupakan *layer* inti dari sistem ini, yaitu sebagai pemroses keseluruhan sistem dan komponen menyimpan semua data sistem. *Layer* ini mempunyai dua jenis server, sebagai berikut :
- a. Local Server : Merupakan *server* utama dari keseluruhan sistem dan digunakan sebagai *server* ketika pengguna berada pada jangkauan *wireless local* yaitu ketika berada di rumah. Platform dari sistem ini adalah Linux(Raspbian), sedangkan perangkat yang digunakan adalah Raspberry Pi.
  - b. *Public Server* dan *Local Server* : Merupakan *server* bantu untuk menghubungkan pengguna dengan *server* lokal. Sehingga pengguna dapat mengakses server ini menggunakan jaringan internet. Kemudian *server* ini menghubungi *server* lokal.
- 4) *Platform Layer* : Sistem ini dibangun berbasis web sehingga pengguna dapat menggunakan *Web Browser* untuk mengakses sistem.
- 5) *User Communication Media Layer* : Pada *layer* ini terdapat dua jenis *layer*, sebagai berikut :
- a. *Wifi* : Jalur komunikasi ini digunakan ketika pengguna berada pada jangkauan *local wifi* yaitu ketika pengguna berada di rumah. Pada jalur

komunikasi ini pengguna tidak perlu terhubung dengan internet. Pengguna cukup mengakses jaringan lokal untuk mengakses sistem.

b. *Internet* : Jalur komunikasi ini digunakan ketika pengguna berada di luar jangkauan *wifi* yaitu ketika pengguna berada di luar rumah. Jenis layanan yang digunakan biasanya GPRS, HSDPA, 3G, 4G dan lain sebagainya.

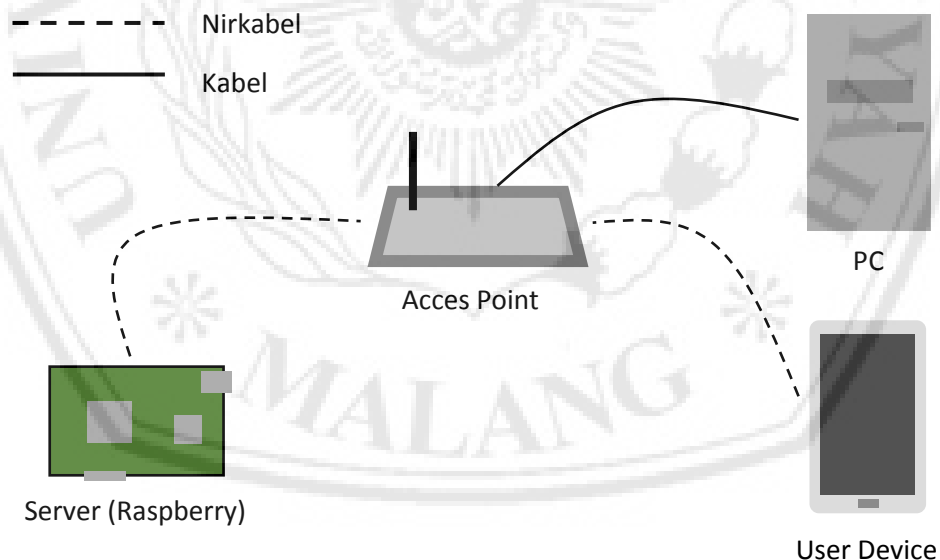
6) *UI Layer* : *Layer* ini berhubungan langsung dengan pengguna. Pengguna dapat mengakses sistem menggunakan *Web Browser*.

### 3.2.2 Komunikasi

Komunikasi pada sistem ini terbagi menjadi 2 bagian yaitu komunikasi dengan *user* dan komunikasi dengan *device clients*. Berikut penjelasannya :

#### 3.2.2.1 Komunikasi dengan User

Komunikasi antara sistem dengan *user* bisa dilakukan di dalam rumah atau bisa dilakukan di luar rumah. Aplikasi pendukung untuk membuka aplikasi adalah *browser* karena aplikasi ini berbasis web. Jika pengguna berada di dalam rumah komunikasi dengan sistem bisa menggunakan media jaringan lokal seperti jalur nirkabel (*wifi*) atau menggunakan kabel UTP. Perhatikan Gambar 3.2.

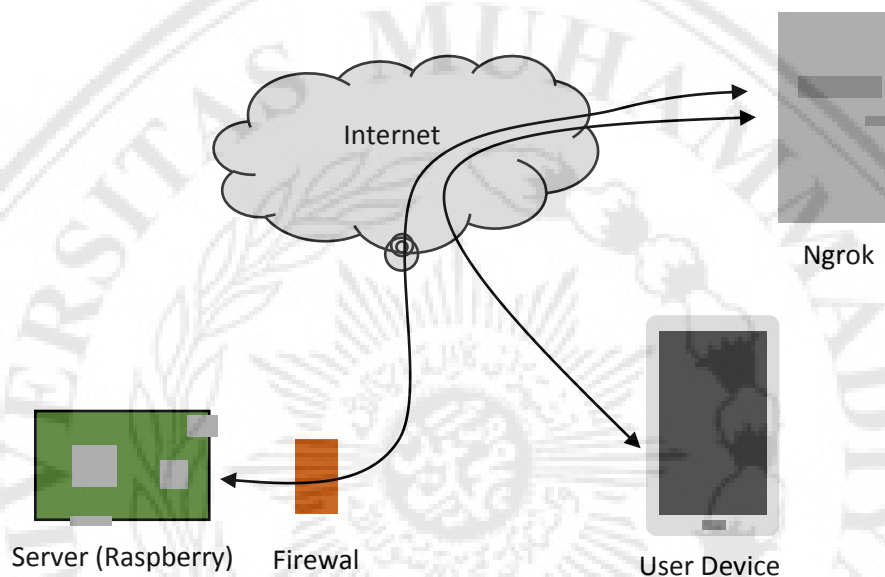


**Gambar 3.2** Topologi jaringan komunikasi dengan *user*

Mengembangkan arsitektur IoT pada salah satu jurnal Seminar Nasional dan Expo Teknik Elektro 2015[11] dan salah satu jurnal SMARTICS [12]. Jika pengguna berada diluar rumah, komunikasi bisa dilakukan dengan mengakses *domain tunneling* yang sudah disediakan dengan syarat *device* harus terhubung

dengan internet. Jalur komunikasi ini digunakan ketika pengguna berada di luar jangkauan *wifi* yaitu ketika pengguna berada di luar rumah. Jenis layanan internet yang biasa digunakan adalah GPRS, HSDPA, 3G, 4G dan internet biasa.

Untuk membuat *server* Raspberry agar bisa diakses oleh pengguna dari internet peneliti menggunakan aplikasi *tunneling* Ngrok. Dengan menggunakan aplikasi *tunneling* Ngrok pengguna dapat mengakses secara langsung *server* lokal dari internet. Perhatikan Gambar 3.3, gambar tersebut mengilustrasikan bagaimana aplikasi *tunneling* Ngrok bekerja.



**Gambar 3.3** Alur komunikasi data pengguna melalui internet

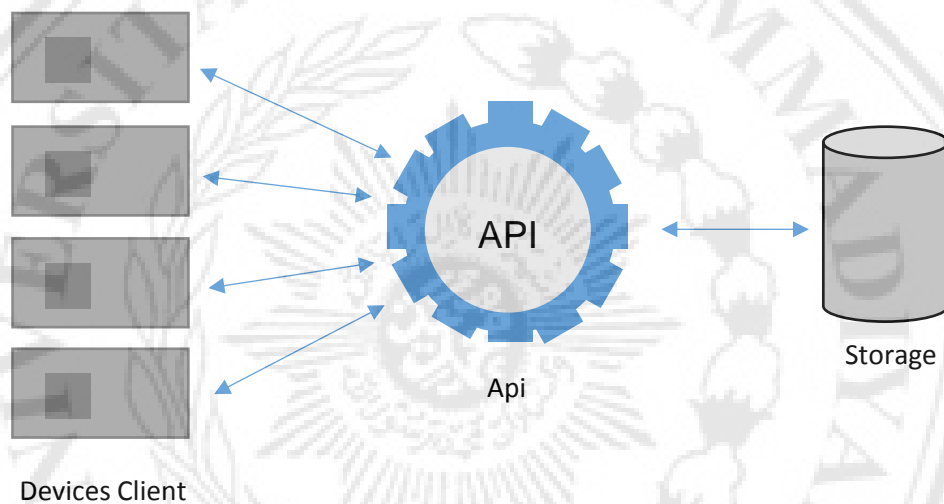
### 3.2.2.2 Komunikasi dengan *Device Clients*

Komunikasi dengan *device clients* dilakukan dengan menggunakan jalur komunikasi *wireless*. Protokol yang digunakan adalah protokol http. *Device clients* akan mengakses *web service*(Api) yang disediakan oleh *server* setiap saat dan berulang sesuai dengan *delay* yang diberikan oleh *server*.

Sistem yang dibuat oleh peneliti merupakan sistem sentral sehingga masih ada sub sistem yang harus terhubung dengan sistem utama. Sub sistem yang lain seperti Door Lock, Lamp Control, Watt Meter & Electricity Control, Camera Watcher dan IR Custom. Untuk menghubungkan sub sistem ini dengan sistem central yang akan dibuat maka akan dibutuhkan model transmisi yang baik seperti model API.

Dengan menggunakan API sub sistem akan mudah dalam melakukan pengiriman data seperti menyimpan data-data sensor oleh *device client* dan pemberian perintah kepada *device client* oleh *server* hanya dengan mengakses URL yang sama untuk semua jenis *device*.

Pengiriman data pada API menggunakan *protocol html* dan data yang dikirim akan ditransformasi ke dalam bentuk *json*. Sehingga *device* yang melakukan *request* cukup mentransformasikan kembali dari bentuk *json* kedalam bentuk *objek* atau *array*. Setiap jenis *device* akan menerima variabel-variabel dalam bentuk *json* yang berbeda-beda, bergantung dengan jenis *device* yang melakukan *request* pada *server*. Perhatikan Gambar 3.4.



**Gambar 3.4** Cara kerja API

Alur komunikasi data satu persatu digambarkan pada Table 3.1.

**Tabel 3.1** Alur komunikasi menggunakan API

<i>Client Device</i>	<b>API</b>
1. Request premision access.	
	2. Membuat <i>session</i> (IP Add. dan Mac Add. <i>device client</i> )
	3. Inisialisasi kode acak (bilangan)
	4. Mengirim kode acak



5. Membuat kode baru sesuai dengan kode acak yang diterima dengan aturan yang sudah disepakati.	
6. Mengirim kode baru.	
	7. Validasi kode dari <i>device client</i> .
	8. Mencari <i>device data</i> di <i>storage</i> sesuai dengan mac <i>device client</i> .
	9. Mengirim data <i>device client</i> sesuai dengan data yang sudah ada.
10. Menyimpan nilai awal dari <i>server</i> dan diterapkan pada <i>sistem device</i> .	
11. Mengirim data sesuai dengan jenis <i>device client</i>	
	12. Menerima data dan disimpan di <i>storage</i> .
	13. Mengambil data status <i>device</i> yang baru
	14. Mengirim data status kembali ke <i>device client</i>
15. Menyimpan nilai awal dan diterapkan pada <i>sistem device</i> .	
16. Mengirim data sesuai dengan jenis <i>device client</i>	
Dan seterusnya sampai koneksi <i>device client</i> dan <i>server</i> terputus. Jika koneksi terputus maka akan mengulang lagi dari langkah pertama.	

### 3.2.3 Efisiensi Energi

Pada penelitian sebelumnya yang dilakukan oleh Manan Mehta pada Jurnal dengan judul *ESP8266: A Breakthrough in Wireless Sensor Networks and Internet*

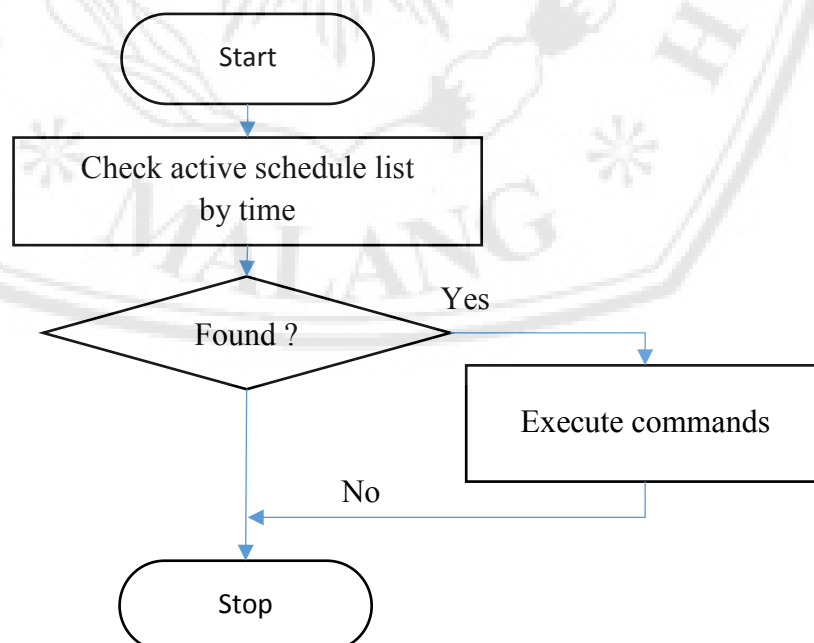
of Things [13] menjelaskan bahwa modul komunikasi ESP8266 terdapat perbedaan konsumsi daya pada setiap *state type*, perbedaannya dapat dilihat pada Tabel 3.2.

**Tabel 3.2** Variasi Daya ESP8266

Type State	Daya
Transmit 802.11b, CCK 1Mbps, POUT=+19.5dBm	215 mA
Receive 802.11b, packet length=1024 byte, -80dBm	60 mA
Standby	0.9 mA
Deep Sleep	10 mA
Power save mode DTIM 1	1.2 mA
Power save mode DTIM 3	0.86 mA

Pada Tabel 3.2 dapat disimpulkan bahwa penggunaan daya paling besar adalah ketika modul melakukan pengiriman data, sehingga semakin banyak data yang dikirim maka konsumsi daya semakin banyak. Peneliti mengusulkan untuk menggunakan *delay* dalam pengiriman data sebagai pembatas dalam penggunaan energi pada sebuah modul, sehingga *delay* pengiriman *device client* dapat diatur berbeda oleh *server* bergantung dengan penggunaannya untuk mengurangi penggunaan daya yang lebih boros.

#### 3.2.4 Scheduling System



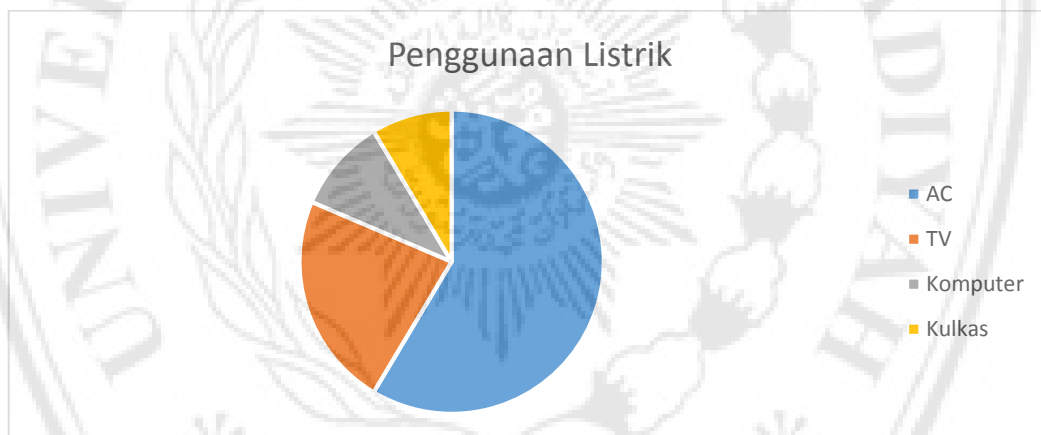
**Gambar 3.5** Flowchat scheduling system pada Php

Pada sistem ini terdapat otomatisasi kontrol yang bekerja dengan terjadwal. Bahasa pemrograman yang digunakan menggunakan bahasa pemrograman Php kemudian *trigger* untuk memprosesnya dilakukan oleh *server* itu sendiri menggunakan bahasa *shell* yang akan berjalan pada setiap *server* menyala.

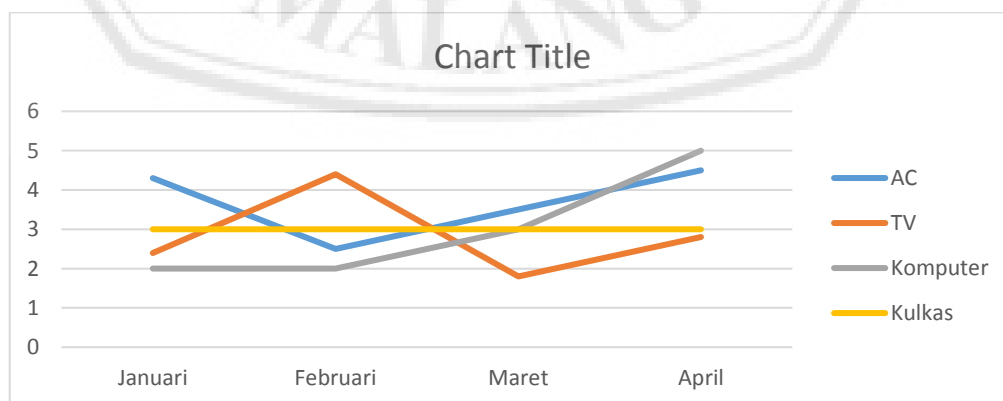
Pada *shell* terdapat perulangan yang tidak ada batasnya yang terdapat perintah untuk mengakses URL. Pada URL tersebut terdapat perintah dalam bentuk pemrograman Php untuk melakukan proses pengecekan dan perintah eksekusi yang terjadwal sesuai dengan data jadwal yang sudah dibuat oleh user. Perhatikan Gambar 3.5.

### 3.2.5 Report

*Report* atau laporan merupakan salah satu pokok dari kegunaan sistem ini. *Report* yang akan dibuat adalah laporan prosentase penggunaan listrik dari setiap *device* dan per satuan *device* dengan skala harian, bulanan, dan tahunan. Perhatikan Gambar 3.6 dan Gambar 3.7

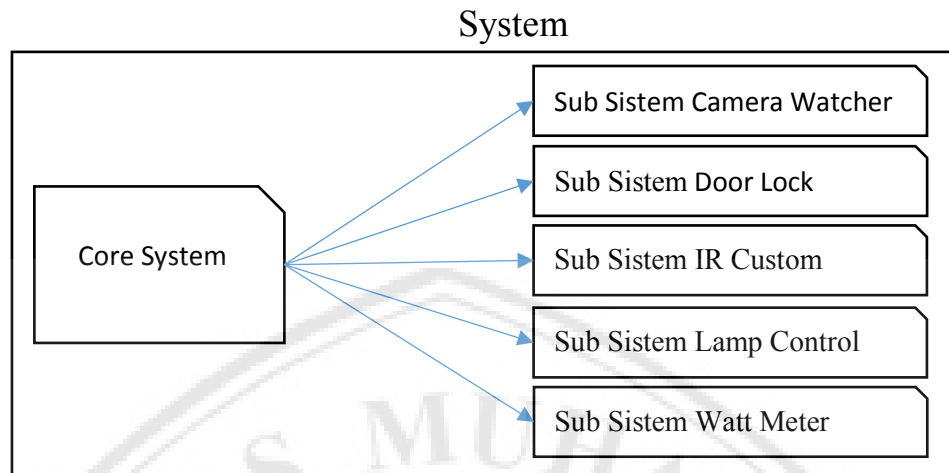


**Gambar 3.6** Desain laporan prosentase penggunaan listrik



**Gambar 3.7** Desain laporan penggunaan listrik setiap bulan

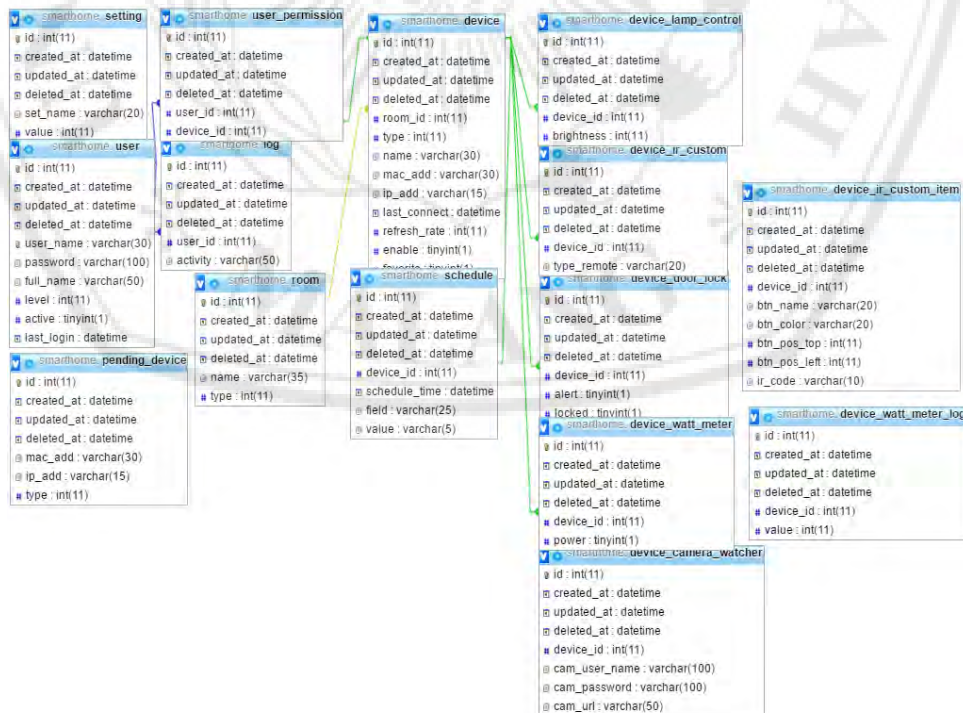
### 3.2.6 Packaging



**Gambar 3.8** Pemetaan modul-modul *devices client*

Pada sistem ini dilakukan *packaging* dengan mengumpulkan bagian-bagian *device clients* sesuai dengan tipenya sehingga jika terjadi pengembangan dengan menambah jenis *device* yang baru maka pengembang berikutnya hanya menambahkan modul *device client* jenis baru yang akan dibuat. Pada sistem ini terdapat 5 modul yaitu Camera, Door Lock, IR Control, Lamp Control dan Watt Meter. Perhatikan Gambar 3.8.

### 3.2.7 Desain Database



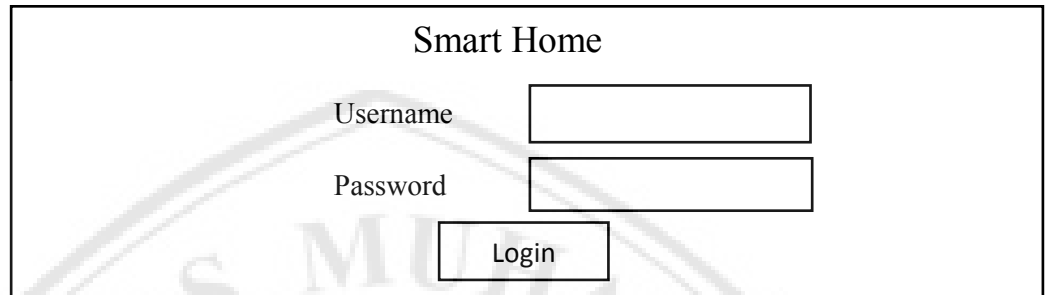
**Gambar 3.9** Desai Database

Adapun desain database dapat dilihat pada Gambar 3.9.

### 3.2.8 User Interface

Berikut desain *user interface* pada setiap halaman yang akan dibuat.

#### 3.2.8.1 Halaman Login



A login form titled "Smart Home". It contains two input fields: "Username" and "Password". Below the "Password" field is a "Login" button.

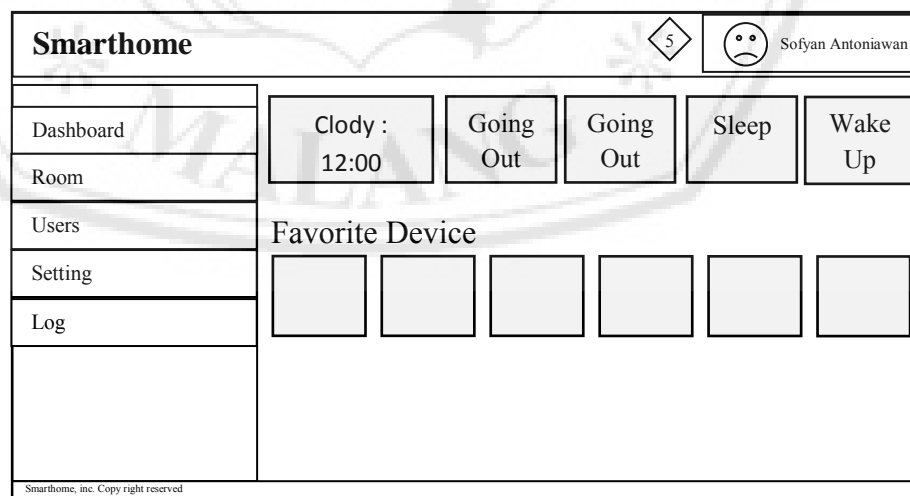
**Gambar 3.10** Desain user interface login

Halaman login merupakan halaman yang harus dilewati oleh semua *user* yaitu dengan memasukkan *username* dan *password*. Jika kedua *parameter* itu *valid* maka *user* dapat masuk ke sistem. Perhatikan Gambar 3.10.

#### 3.2.8.2 Halaman Dashboard

Halaman *dashboard* merupakan halaman *admin* pertama yang diakses setelah *user* ketika berhasil *login* (Gambar 3.11). Halaman ini berisi komponen sebagai berikut :

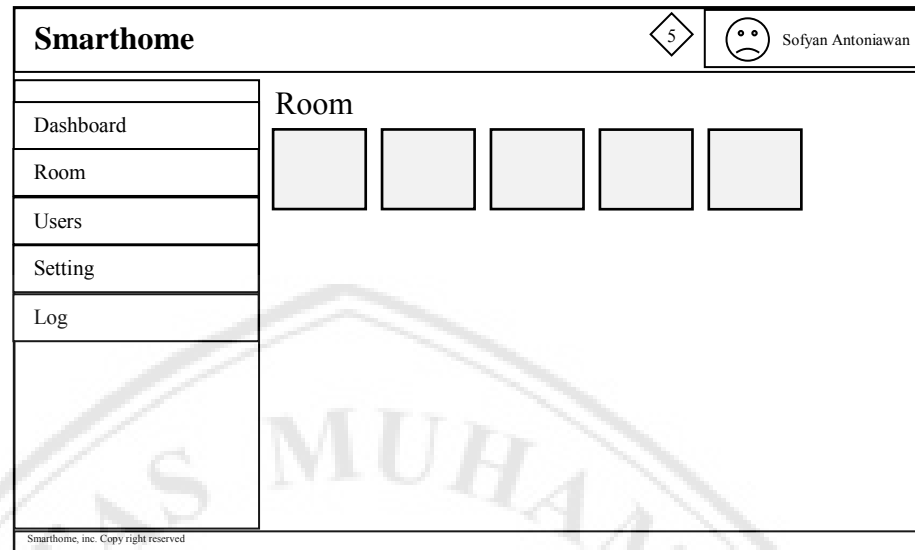
1. Jam sekarang beserta ramalan cuaca dari Yahoo Weather
2. *Shortcut* : “Going Out”, “Going Out”, “Sleep” dan “Wake Up”.
3. Daftar *device favorite* yang berisi *devices* yang dianggap *favorite*.



A dashboard titled "Smarthome". It features a sidebar on the left with a menu containing "Dashboard", "Room", "Users", "Setting", and "Log". The main content area includes a header with a diamond icon containing the number "5" and a user profile icon with the name "Sofyan Antoniaawan". Below the header, there are five buttons: "Clody : 12:00", "Going Out", "Going Out", "Sleep", and "Wake Up". Underneath these buttons is a section titled "Favorite Device" with six empty square boxes. At the bottom left, there is a small copyright notice: "Smarthome, inc. Copy right reserved".

**Gambar 3.11** Halaman *Dashboard*

### 3.2.8.3 Halaman *Room*

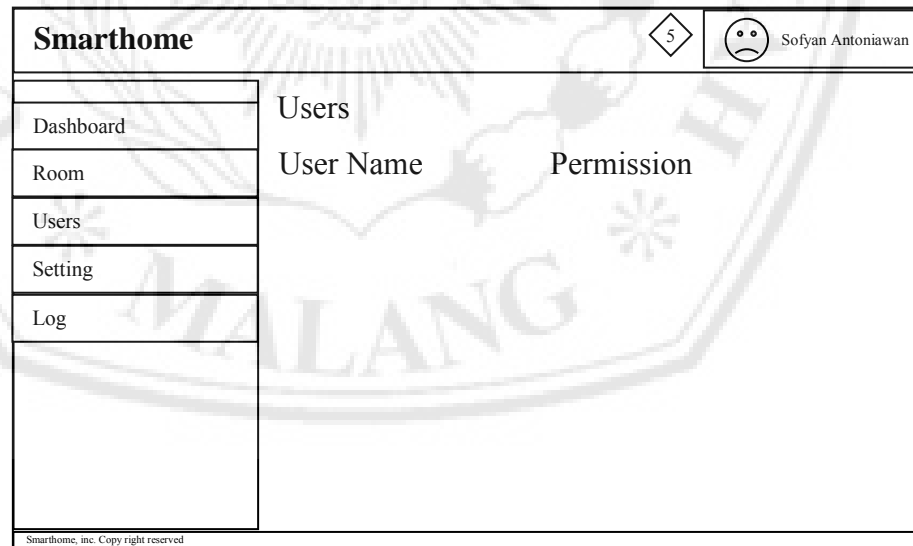


**Gambar 3.12** Halaman *Room*

Layout ini berisi daftar ruangan yang sudah dibuat, ketika user memilih ruangan maka akan menampilkan daftar perangkat yang ada di dalam ruangan. Perhatikan Gambar 3.12.

### 3.2.8.4 Halaman *User*

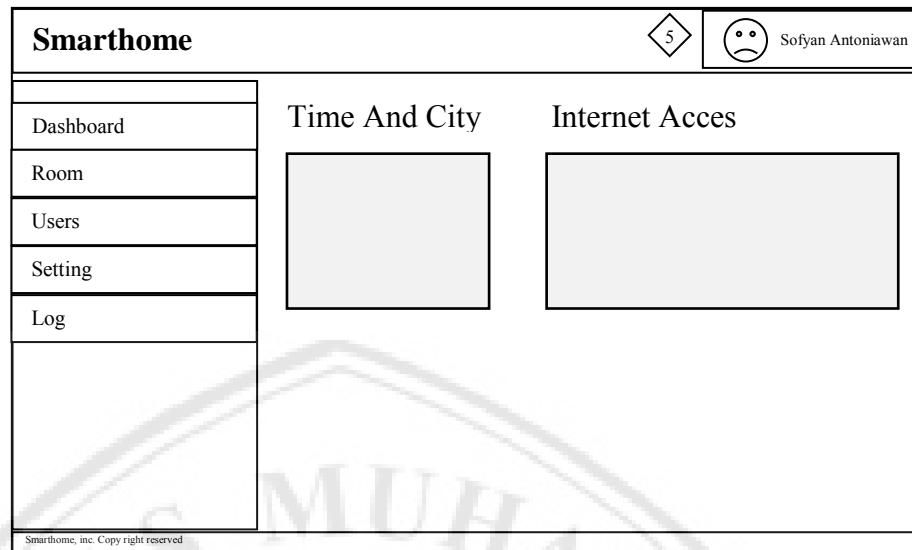
Berisi daftar *user* yang sudah ada dan daftar *device client* yang diberi otorisasi pada *user* pada tiap-tiap *user*. Perhatikan Gambar 3.13



**Gambar 3.13** Halaman *Dashboard*

### 3.2.8.5 Halaman *Setting*

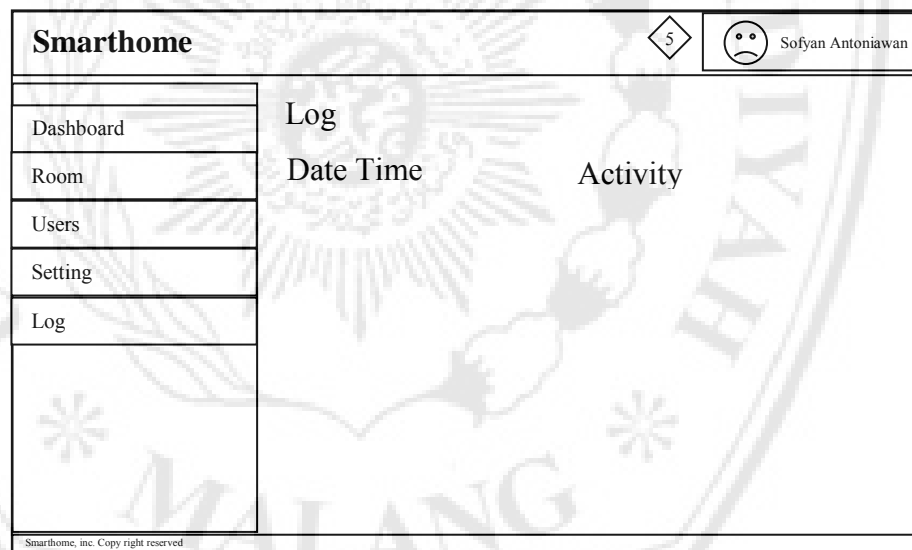
Berisi pengaturan sistem seperti pengaturan jam, pengaturan kota sekarang dan pengaturan akses *server* dari internet. Perhatikan pada Gambar 3.14.



**Gambar 3.14** Halaman *Setting*

### 3.2.8.6 Halaman *Log*

Berisi beberapa kegiatan-kegiatan *user* seperti mematikan lampu dan menyalakan lampu. Perhatikan pada Gambar 3.15.



**Gambar 3.15** Halaman *Log*